**Bitcoin: A Peer-to-Peer Electronic Cash System**

This paper focuses on a peer-to-peer version of electronic cash that would allow payments to be sent directly from one party to another without going through a financial institution and establishes the foundation of the technology used to achieve this objective i.e. Blockchain.

Today, transactions on the internet rely on financial institutions to process payments. The need for trusted third parties to mediate transactions makes non-reversible transactions impossible and increases the transaction costs. There is an acceptance that some fraud is inevitable. There is a need for an electronic payment system that uses cryptographic proof rather than trust and enables parties to transact with each other bypassing central third party.

A Bitcoin is a chain of digital signatures. Every owner of an electronic coin passes it to the next owner by digitally signing:

* A hash of the previous transaction
* The public key of the new owner
* Adding above 2 components to the end of the electronic coin

A payee can verify the signatures to verify the chain of ownership. To avoid double spending of the coin without a trusted third party requires that transactions are declared publicly and all participants agree on a single history of the order in which they were received.

The timestamp server takes the hash of a block of items, timestamps them and publicly publishes the hash. Each timestamp includes the previous timestamp, creating a chain, and as new timestamp hashes are added the chronological order and links are strengthened. Implementing a distributed time-stamp server requires a proof-of-work system.

Proof-of-work requires scanning for a value which when hashed (e.g. using SHA-256) the hash value starts with a number of zero-bits. The average work required is exponential in the number of zero bits required. It Ensures that a verified block cannot be changed because all later blocks that are chained to it will also need to be changed (each subsequent block would need to be verified, requiring increasing CPU effort). It is based on a one-CPU-one-vote system, ensuring that the majority decision is based on the longest chain which requires the most proof-of-work effort.

The steps to run the network are as follows:

* New transactions are broadcast to all nodes.
* Each node collects new transactions into a block.
* Each node works on finding a difficult proof-of-work for its block.
* When a node proof-of-work, it broadcasts the block to all nodes.
* Nodes accept the block only if all transactions in it are valid and not already spent.
* Nodes express their acceptance of the block by working on creating the next bloc in the chain, using the hash of the accepted block as the previous block.

The goal in peer-to-peer electronic cash system is to encourage nodes to connect to the network and validate transactions. The first block in a transaction starts a new coin which is owned by

the creator of the block. In order to generate new blocks, and therefore coins (value), CPU and electricity are needed. If the output of a transaction is less than the input value, a transaction fee is added to the block containing the transaction.

Any old transactions can be removed to save disk space. To enable this removal without breaking the block hash value, transactions are hashed in a Merkle Tree. This allows for old blocks to be consolidated by essentially removing the tree branches, but keeping the root.

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he’s convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it’s timestamped in. He can’t check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. The verification is reliable as long as honest nodes control the network. To allow value to be divided and merged, transactions contain various inputs and outputs. For example, a single input from a large transaction, or many smaller inputs.

Traditional banking limits access to information to just those involved in the transaction and the trusted third party. This is not workable in a model where the transactions are broadcast publicly, but the need for privacy is still important. Privacy is maintained by keeping public keys anonymous. A transfer can happen without knowing who is involved in the transaction.

Where honest nodes control the majority of CPU power, a peer-to-peer network that uses proof-of-work to record public transactions makes it computationally impractical for attackers to tamper with.